Proof of Concept:

CFFDRS Wildfire Fuel Mapping Prototype

CabinGIS

24/02/2022

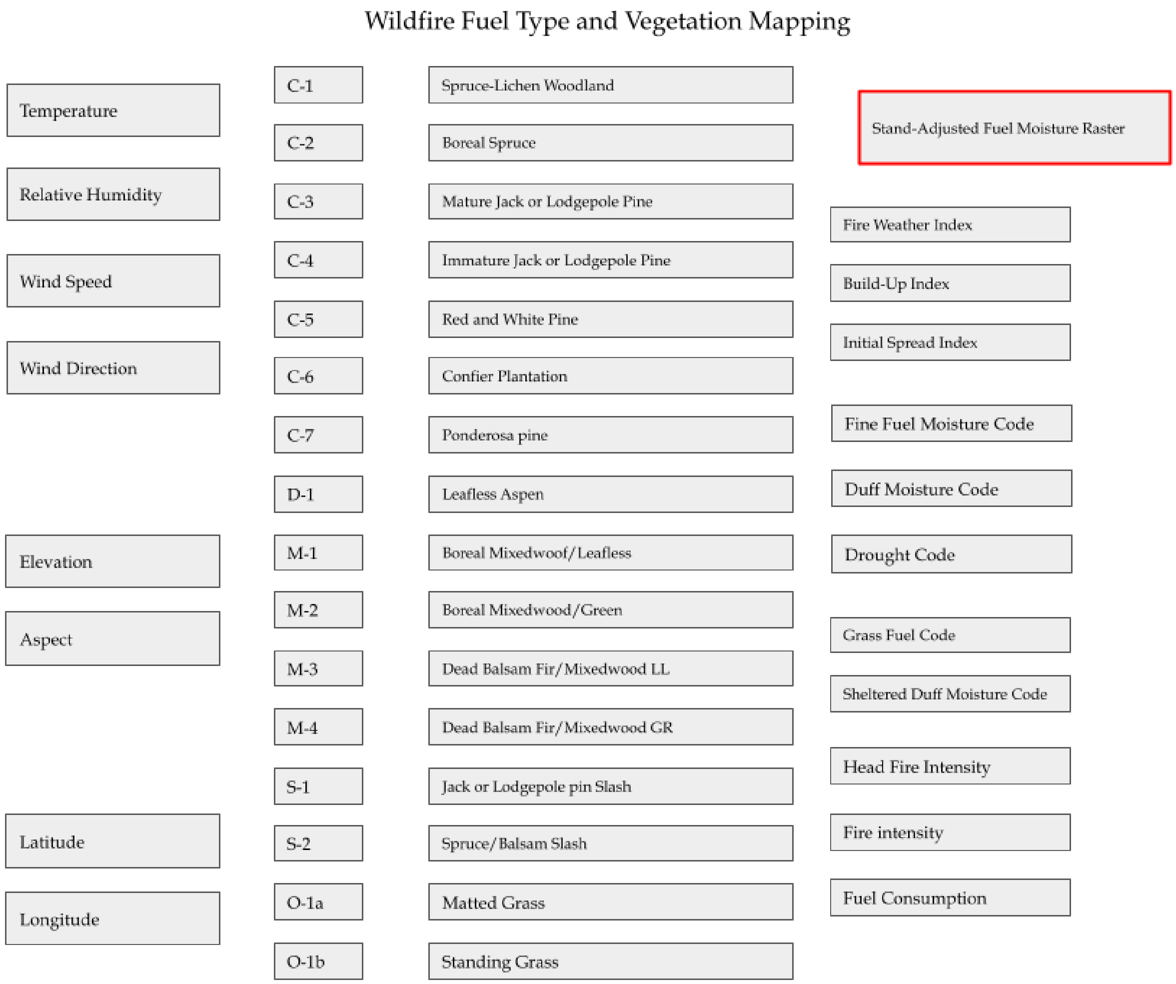
Contents

Action

Building on the momentum and ideas of our last meeting, the following pipeline was attempted to produce the wildfire fuel mapping outputs described in the NRC grant "High-Resolution Mapping":

- NRC Grant: https://www.ic.gc.ca/eic/site/101.nsf/eng/00157.html

Using the new cffdrs R-package (Wang et al. [2017]; Van Wagner and Pickett [1985]; Van Wagner [1987]), we developed rasters of forest fuel moisture code and wildfire weather indices (Table 1) that were used to fit the stand-adjusted fine-fuel model (Wotton and Beverly [2007]) applied to vegetation rasters classified according to 16 fuel classes of the BC forest fuel typing algorithm (Perrakis et al. [2018]). Resulting rasters were then used to fit the Canadian Forest Fire Behaviour Prediction model from which we drafted raster maps representing Head Fire Index (HFI) and Fire Intensity maps (FI) (Figure 1) for the Peachland watershed administration area on the day of June 30th 2021.

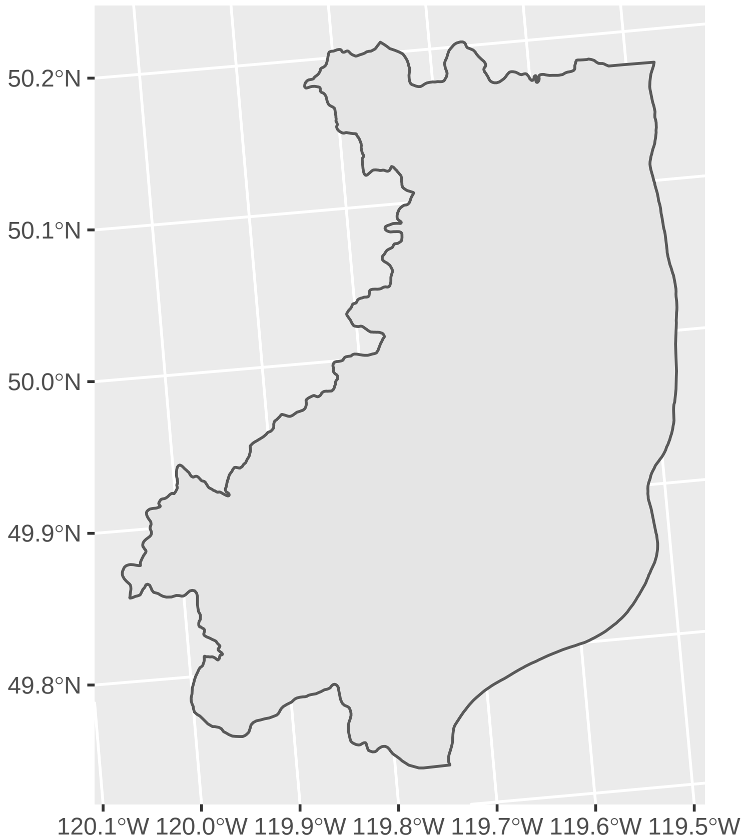


1 BC WildFire Fuel Typing VRI-Layer Algorithm

1.1 Import Site Data

An area-of-interest polygon was adopted from the Peachland watershed administration boundary, which was downloaded using the ImapBC custom warehouse download tool. This outline was used to clip and reproject all other raster objects and output using the GeoBC accepted BCAlbers CRS (EPSG:3005) and ggplot functions seen below.

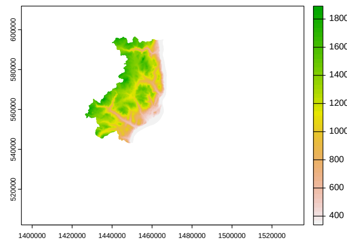
```
watershed_bdrys = read_sf("./Data/BCGW_7113060B_1645786298548_3276/LWADM_WATMGMT_PREC_AREA_SVW/LWADM_PA
peachland = watershed_bdrys[watershed_bdrys$PRECNC_NAM == "Peachland", ]
st_crs(peachland) = "epsg:3005"
ggplot(peachland) +
  geom_sf() +
  coord_sf()
```

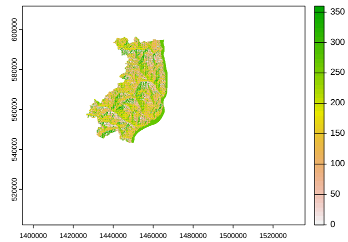


1.2 Import Topographical Data

For testing purposes, we acquired open source LiDAR data using the 'elevatr' package to download the SRTM 3arc-seconds dataset (Van Zyl [2001]). To land the download, we generated a spatial data object from the peachland simple feature. All rasters were labelled to match those required by the fwiRaster and fpbRaster functions. The 3-arc second DEM was transformed into a spatRaster for speedier operations and disaggreated from 98m resolution to 32m~ resolution. Slope and aspect rasters calculated using the terra::terrain function and clipped to the preachland watershed boundary.

```
peachland_sp = as(peachland, "Spatial")
ELV = get_elev_raster(peachland_sp, z = 9)
ELV = rast(ELV)
ELV = disagg(ELV, fact=3.3)
crs(ELV) = "epsg:3005"
ELV = mask(ELV, vect(peachland_sp))
GS = terrain(ELV, "slope")
Aspect = terrain(ELV, "aspect")
GS = mask(GS, vect(peachland_sp))
Aspect = mask(Aspect, vect(peachland_sp))
plot(GS)
plot(ELV)
plot(Aspect)
```



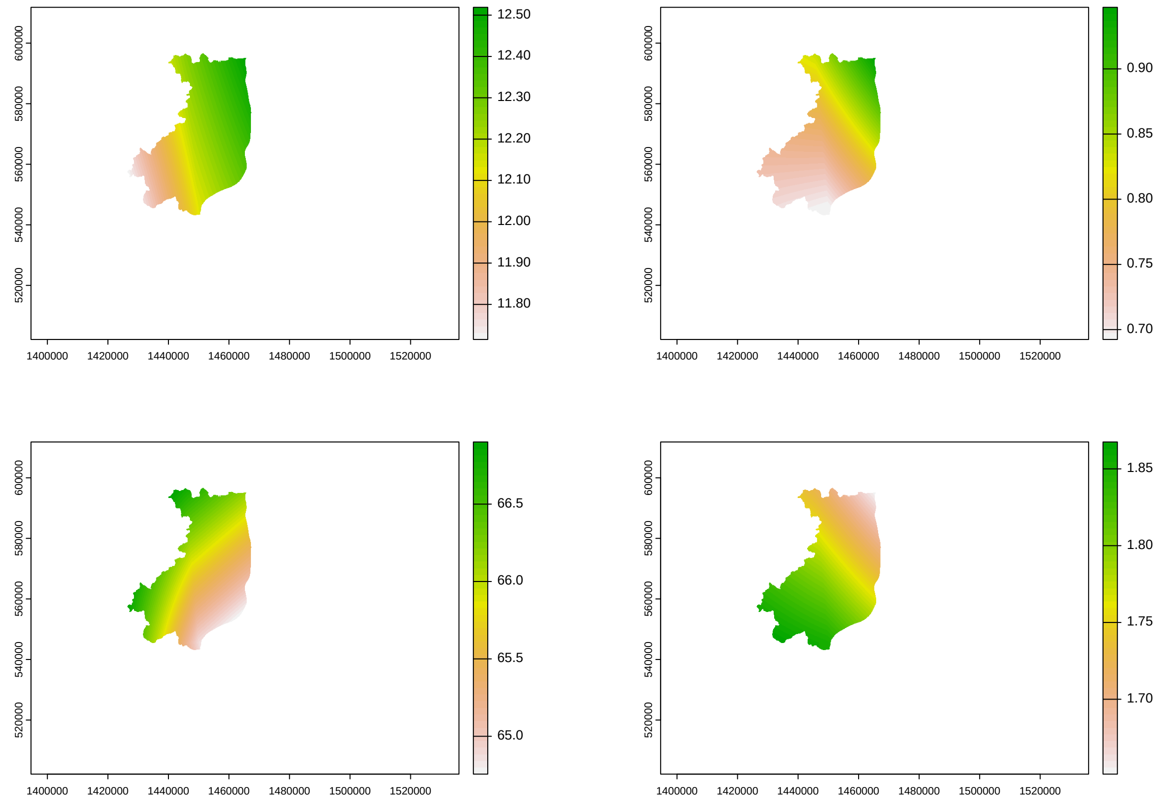


1.3 Import Climate Data

Climate variables were downloaded as NetCDF files from NASA Power platform and read directly into R as rasters of 1) mean daily temperature at 2m, 2) mean daily precipitation, 3) mean relative humidity, 4) and mean wind speed at 10m. The NASA Power platform supports some very user-friendly API links for static data sources that might be useful for this proposed grant project. REminder tho, some API's prefer dealing with dataframe inputs so might be worht preparing df pipe while still fresh in the head.

- NASA Power Platform: https://power.larc.nasa.gov/data-access-viewer/

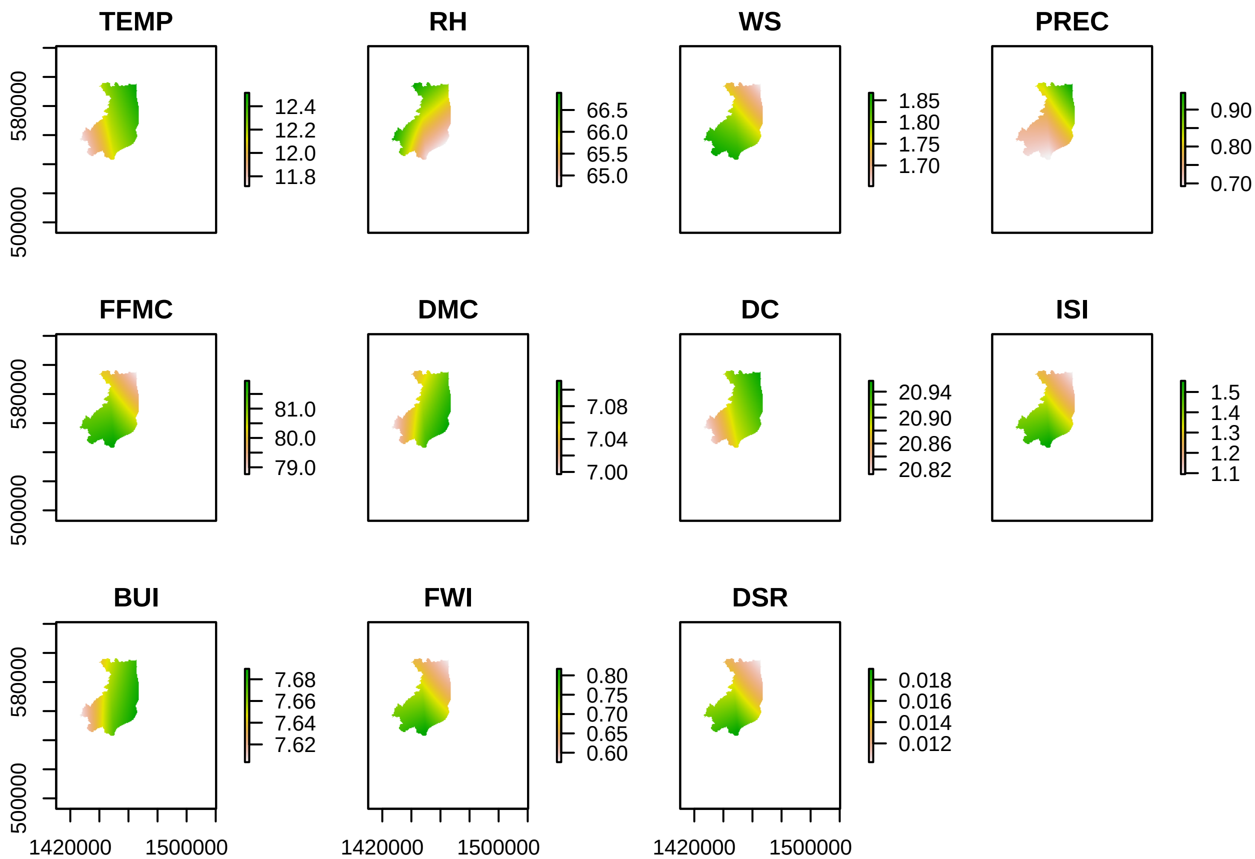
```
temp = terra::rast("./Data/temp.nc")
prec = terra::rast("./Data/prec.nc")
rh = terra::rast("./Data/rh.nc")
ws = terra::rast("./Data/ws.nc")
temp = terra::resample(temp, ELV, method="bilinear")
prec = terra::resample(prec, ELV, method="bilinear")
rh = terra::resample(rh, ELV, method="bilinear")
ws = terra::resample(ws, ELV, method="bilinear")
temp = mask(temp, vect(peachland))
prec = mask(prec, vect(peachland))
rh = mask(rh, vect(peachland))
ws = mask(ws, vect(peachland))
temp = mean(temp)
prec = mean(prec)
rh = mean(rh)
ws = mean(ws)
names(temp) = 'temp'
names(prec) = 'prec'
names(rh) = 'rh'
names(ws) = 'ws'
plot(temp)
plot(prec)
plot(rh)
plot(ws)
```



1.4 Derive CFFDRS Wildfire Weather Rasters

Interpolated climate predictors were assembled as a raster stack and inputted into the fwiRaster function. The 'out="all"' option was selectedin the fwiRaster function which gave us the additional raster outputs for Initial Spread Index (isi), and Build-up Index (bui).

```
temp = raster::raster(temp)
prec = raster::raster(prec)
rh = raster::raster(rh)
ws = raster::raster(ws)
stack = stack(temp, rh, ws, prec)
fwi_outputs = fwiRaster(stack, out = "all")
plot(fwi_outputs)
```




```
ffmc = raster(fwi_outputs, layer=5)
dmc = raster(fwi_outputs, layer=6)
dc = raster(fwi_outputs, layer=7)
isi = raster(fwi_outputs, layer=8)
bui = raster(fwi_outputs, layer=9)
fwi = raster(fwi_outputs, layer=10)
dsr = raster(fwi_outputs, layer=11)
```

2 Import VRI Data

We imported the VRI dataset from imapBC and imported as a shapefile.shp into R. The BRI object was transformed into a simple feature and processed using sf and dplyr functions. For an idea of CFFDRS data requirements, we've taken a look at the in-built sample datasets 'test_fwi' and 'test_fpb' below:

```
library(cffdrs)
print(as_tibble(test_fwi), n = 10)
```

```
## # A tibble: 48 x 9
##     long   lat    yr   mon   day  temp    rh    ws  prec
##    <int> <int> <int> <int> <int> <dbl> <int> <int> <dbl>
##  1  -100    40  1985     4    13  17      42    25   0
##  2  -100    40  1985     4    14  20      21    25   2.4
##  3  -100    40  1985     4    15   8.5    40    17   0
```

```
## 4 -100 40 1985 4 16 6.5 25 6 0
## 5 -100 40 1985 4 17 13 34 24 0
## 6 -100 40 1985 4 18 6 40 22 0.4
## 7 -100 40 1985 4 19 5.5 52 6 0
## 8 -100 40 1985 4 20 8.5 46 16 0
## 9 -100 40 1985 4 21 9.5 54 20 0
## 10 -100 40 1985 4 22 7 93 14 9
## # ... with 38 more rows
```

```r
print(as_tibble(test_fbp), n = 10)
```

```
## # A tibble: 20 x 24
##       id FuelType   LAT  LONG   ELV  FFMC   BUI    WS    WD    GS    Dj    D0
##    <int> <fct>    <int> <int> <int> <dbl> <int> <dbl> <int> <int> <int> <int>
##  1     1 C-1         55   110    NA  90     130  20       0    15   182    NA
##  2     2 C2          50    90    NA  97     119  20.4     0    75   121    NA
##  3     3 C-3         55   110    NA  95      30  50       0     0   182    NA
##  4     4 C-4         55   105   200  85      82   0      NA    75   182    NA
##  5     5 c5          55   105    NA  88      56   3.4     0    23   152   145
##  6     6 C-6         55   105    NA  94      56  25       0    10   152   132
##  7     7 C-7         50   125    NA  88.8    15  22.1   270    15   152    NA
##  8     8 D-1         45   100    NA  98     100  50     270    35   152    NA
##  9     9 M-1         47    85    NA  90      40  15.5   180    25   182    NA
## 10    10 M-2         63   120   100  97     150  41     180    50   213    NA
## # ... with 10 more rows, and 12 more variables: hr <dbl>, PC <int>, PDF <int>,
## #   GFL <dbl>, cc <int>, theta <int>, Accel <int>, Aspect <int>, BUIEff <int>,
## #   CBH <lgl>, CFL <lgl>, ISI <int>
```

3 Import VRI Data

```r
vri_sf = read_sf("./Data/BCGW_7113060B_1645786298548_3276/VEG_COMP_LYR_R1_POLY/VEG_R1_PLY_polygon.shp")
vri_sf = st_intersection(st_make_valid(vri_sf), peachland)
stand = vri_sf["SPEC_CD_1"] %>% mutate(SPEC_CD_1 = as.factor(SPEC_CD_1))
stand = rename(stand, stand = SPEC_CD_1)
summary.factor(stand$stand)
```

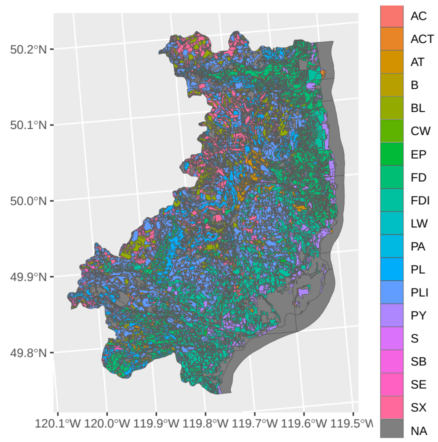
```
##   AC  ACT   AT    B   BL   CW   EP   FD  FDI   LW   PA   PL  PLI   PY    S   SB
##    1   33  496    4 1122    9  104 2255 1648   20   10 1804 1752  299   60    1
##   SE   SX NA's
##  214 1349  875
```

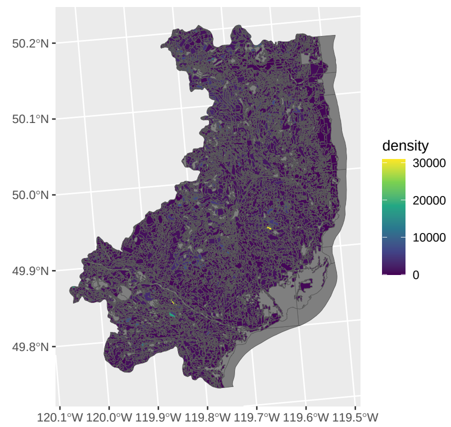
```r
density = vri_sf["LIVE_STEMS"] %>% mutate(LIVE_STEMS = as.numeric(LIVE_STEMS))
density = rename(density, density = LIVE_STEMS)
summary(density)
```

```
##     density                    geometry
##  Min.   :    0.0   MULTIPOLYGON :   98
##  1st Qu.:  413.0   POLYGON      :11958
##  Median :  740.0   epsg:3005    :    0
```

```
## Mean   :  982.8   +proj=aea ...:    0
## 3rd Qu.: 1116.0
## Max.   :30820.0
## NA's   :812
```

```r
ggplot(stand) + geom_sf(aes(fill=stand), size = 0.05)
ggplot(density) + geom_sf(aes(fill=density), size = 0.0005) + scale_fill_viridis_c()
```






```r
# EDA scopes
fields::stats(vri_sf$HRVSTDT)
fields::stats(vri_sf$C_I_CODE)
fields::stats(vri_sf$BEC_ZONE)
fields::stats(vri_sf$BCLCS_LV_1)

psych::describeData(vri_sf)



summary.factor(vri_sf$SPEC_CD_1)

C1_fuel = vri_sf %>%
  dplyr::filter(BCLCS_LV_2 == 'V', BCLCS_LV_1 == 'V',
              HRVSTDT > 19970000, HRVSTDT > 20140000,
              BEC_ZONE == "BWBS" | BEC_ZONE == "SWB",
              SPEC_CD_1 =='S' | SPEC_CD_1 == 'SB' |
                SPEC_CD_1 == 'SE' | SPEC_CD_1 =='SX')

C2_fuel = vri_sf %>%
  dplyr::filter(BCLCS_LV_2 == 'V', BCLCS_LV_1 == 'V',
              HRVSTDT > 19970000, HRVSTDT > 20140000,
              BEC_ZONE == "BWBS" | BEC_ZONE == "SWB",
              SPEC_CD_1 =='S' | SPEC_CD_1 == 'SB' |
                SPEC_CD_1 == 'SE' | SPEC_CD_1 =='SX')

C3_fuel = vri_sf %>%
  dplyr::filter(BCLCS_LV_2 == 'V', BCLCS_LV_1 == 'V',
              HRVSTDT > 19970000, HRVSTDT > 20140000,
              SPEC_CD_1 =='Pl' | SPEC_CD_1 == 'Pli' |
                SPEC_CD_1 == 'Plc' | SPEC_CD_1 =='Pj'|
```

```
                  SPEC_CD_1 == 'P' | SPEC_CD_1 =='SE',)


C4_fuel = vri_sf %>%
  dplyr::filter(BCLCS_LV_2 == 'V', BCLCS_LV_1 == 'V',
                HRVSTDT > 19970000, HRVSTDT > 20140000,
                SPEC_CD_1 =='Pl' | SPEC_CD_1 == 'Pli' |
                  SPEC_CD_1 == 'Plc' | SPEC_CD_1 =='Pj'|
                  SPEC_CD_1 == 'P' | SPEC_CD_1 =='SE',)


C4_fuel = vri_sf %>%
  dplyr::filter(BCLCS_LV_2 == 'V', BCLCS_LV_1 == 'V',
                HRVSTDT > 19970000, HRVSTDT > 20140000,
                SPEC_CD_1 =='Pl' | SPEC_CD_1 == 'Pli' |
                  SPEC_CD_1 == 'Plc' | SPEC_CD_1 =='Pj'|
                  SPEC_CD_1 == 'P', vri_sf$PROJ_AGE_1 < 2)
# Competitor Tools for Fuel Typing in BC 2022

#### *fwiRaster and sdmc calculated based on daily climate records*

#### *gfmc and hffmc calculated based on hourly climate records - key to CFFDRSv2.0*

#### *Start date of fire season calculated with fireSeason*

#### *All outputs generated for once-daily calcuylations for the full fireSeason chronologically using
```

References

Daniel DB Perrakis, George Eade, and Dana Hicks. *British Columbia wildfire fuel typing and fuel type layer description*. Canadian Forest Service, Natural Resources Canada, 2018.

CE Van Wagner. Development and structure of the canadian forest fire weather index system. canadian forestry service forestry. Technical report, Technical Report 35, Ottawa, 1987.

CE Van Wagner and TL Pickett. *Equations and FORTRAN program for the Canadian forest fire weather index system*, volume 33. 1985.

Jakob J Van Zyl. The shuttle radar topography mission (srtm): a breakthrough in remote sensing of topography. *Acta Astronautica*, 48(5-12):559–565, 2001.

Xianli Wang, B Mike Wotton, Alan S Cantin, Marc-André Parisien, Kerry Anderson, Brett Moore, and Mike D Flannigan. cffdrs: an r package for the canadian forest fire danger rating system. *Ecological Processes*, 6(1):1–11, 2017.

B Mike Wotton and Jennifer L Beverly. Stand-specific litter moisture content calibrations for the canadian fine fuel moisture code. *International Journal of Wildland Fire*, 16(4):463–472, 2007.